

```

VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLL      IIIIIIIII  000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLL      IIIIIIIII  000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLL      IIIIIIIII  000000000000
VVV      VVV  MMMMMM  MMMMMM  SSS      LLL      III      000      000
VVV      VVV  MMMMMM  MMMMMM  SSS      LLL      III      000      000
VVV      VVV  MMMMMM  MMMMMM  SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSSSSSSSSS  LLL      III      000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSS  LLL      III      000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSS  LLL      III      000000000000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSSSSSSSSSSS  LLLLLLLLLLLLLLLLL  IIIIIIIII  000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSSS  LLLLLLLLLLLLLLLLL  IIIIIIIII  000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSSS  LLLLLLLLLLLLLLLLL  IIIIIIIII  000000000000

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II         SS
LL      II         SS
LL      II         SS
LL      II         SS
LL      II         SSSSSS
LL      II         SSSSSS
LL      II         SS
LL      II         SS
LL      II         SS
LL      II         SS
LL      II         SSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLL.LLLLLL      IIIIII      SSSSSSSS

```

(2)	56	FMG\$MATCH_NAME, general wild card matching
-----	----	---

MATCHNAME
V04-000

Match General Wild Card Specification

B 10

16-SEP-1984 02:18:58 VAX/VMS Macro V04-00
5-SEP-1984 04:41:15 [VMSLIB.SRC]MATCHNAME.MAR;1

Page 1
(1)

OP

```
0000 1      .TITLE  MATCHNAME      Match General Wild Card Specification
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5      *****
0000 6      *
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10     *
0000 11     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     *  TRANSFERRED.
0000 17     *
0000 18     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     *  CORPORATION.
0000 21     *
0000 22     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     *
0000 25     *
0000 26     *  *****
0000 27
0000 28     ++
0000 29
0000 30     FACILITY:  Files-11 Structure Level 2
0000 31
0000 32     ABSTRACT:
0000 33
0000 34         This routine performs the general embedded wild card matching
0000 35         algorithm.
0000 36
0000 37     ENVIRONMENT:
0000 38
0000 39         VAX/VMS Operating System
0000 40
0000 41     --
0000 42
0000 43
0000 44     AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  10-Aug-1979  11:36
0000 45
0000 46     MODIFIED BY:
0000 47
0000 48         V03-001 ACG0378      Andrew C. Goldstein,  6-Dec-1983  16:10
0000 49         Incorporate into system build library
0000 50
0000 51         V02-001 MLJ0031      Martin L. Jack, 4-Aug-1981  6:32
0000 52         Reorganize for simplicity and speed.
0000 53
0000 54     **
```

```
0000 56 .SBTTL FMG$MATCH_NAME, general wild card matching
0000 57
0000 58 :++
0000 59
0000 60 Functional Description:
0000 61 This routine performs the general embedded wild card matching
0000 62 algorithm.
0000 63
0000 64 Calling Sequence:
0000 65 JSB
0000 66
0000 67 Input Parameters:
0000 68 R2 = Length of candidate string.
0000 69 R3 = Address of candidate string.
0000 70 R4 = Length of pattern string.
0000 71 R5 = Address of pattern string.
0000 72
0000 73 Implicit Inputs:
0000 74 none
0000 75
0000 76 Output Parameters:
0000 77 none
0000 78
0000 79 Implicit Outputs:
0000 80 none
0000 81
0000 82 Routines Called:
0000 83 none
0000 84
0000 85 Routine Value:
0000 86 True if the strings match.
0000 87
0000 88 Signals:
0000 89 none
0000 90
0000 91 Side Effects:
0000 92 R1-R5 destroyed.
0000 93
0000 94 :--
0000 95
00000000 96 .PSECT _LIB$CODE,NOWRT,EXE,PIC,SHR
0000 97
0000 98 FMG$MATCH_NAME::
03C0 8F BB 0000 99 PUSH R6,R7,R8,R9 ; Save registers
50 D4 0004 100 CLRL R0 ; Assume failure
56 D4 0006 101 CLRL R6 ; Clear saved candidate count
0008 102
0008 103 : Main scanning loop.
0008 104
54 D7 0008 105 10$: DECL R4 ; Pattern exhausted?
24 19 000A 106 BLSS 30$ ; Branch if yes
51 85 9A 000C 107 MOVZBL (R5)+,R1 ; Get next character in pattern
2A 51 91 000F 108 CMPB R1,#^A'^* ; Pattern specifies wild string?
28 13 0012 109 BEQL 60$ ; Branch if yes
52 D7 0014 110 DECL R2 ; Candidate exhausted?
1F 19 0016 111 BLSS 50$ ; Branch if yes
83 51 91 0018 112 CMPB R1,(R3)+ ; Compare pattern to candidate
```



```
25 EB 13 001B 113 BEQL 10$ ; Branch if pattern equals candidate
    51 91 001D 114 CMPB R1,#^A'X' ; Pattern specifies wild character?
    E6 13 0020 115 BEQL 10$ ; Branch if yes
        0022 116 ;
        0022 117 ; We have detected a mismatch, or we are out of pattern while there is
        0022 118 ; candidate left. Back up to the last '*', advance a candidate character,
        0022 119 ; and try again.
        0022 120 ;
    56 D7 0022 121 20$: DECL R6 ; Count a saved candidate character
    11 19 0024 122 BLSS 50$ ; Branch if no saved candidate
    57 D6 0026 123 INCL R7 ; Set to try next character
    52 56 7D 0028 124 MOVQ R6,R2 ; Restore descriptors to backup point
    54 58 7D 002B 125 MOVQ R8,R4 ;
    D8 11 002E 126 BRB 10$ ; Continue testing
        0030 127 ;
        0030 128 ; Here when pattern is exhausted.
        0030 129 ;
    52 D5 0030 130 30$: TSTL R2 ; Candidate exhausted?
    EE 12 0032 131 BNEQ 20$ ; Branch if no
        0034 132 ;
        0034 133 ; Here to return.
        0034 134 ;
    50 01 D0 0034 135 40$: MOVL #1,R0 ; Set success return
    03C0 8F BA 0037 136 50$: POPR #^M<R6,R7,R8,R9> ; Restore registers
    05 003B 137 RSB ; Return
        003C 138 ;
        003C 139 ; We have detected a '*' in the pattern. Save the pointers for backtracking.
        003C 140 ;
    54 D5 003C 141 60$: TSTL R4 ; Pattern null after '*'?
    F4 13 003E 142 BEQL 40$ ; Branch if yes
    56 52 7D 0040 143 MOVQ R2,R6 ; Save descriptors of both strings
    58 54 7D 0043 144 MOVQ R4,R8 ;
    C0 11 0046 145 BRB 10$ ; Continue testing
        0048 146
        0048 147 .END
```

MATCHNAME
Symbol table

Match General Wild Card Specification

E 10

16-SEP-1984 02:18:58
5-SEP-1984 04:41:15

VAX/VMS Macro V04-00
[VMSLIB.SRC]MATCHNAME.MAR;1

Page 4
(2)

FMG\$MATCH_NAME 00000000 RG 01

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_LIB\$CODE	00000048 (72.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.08	00:00:00.61
Command processing	130	00:00:00.46	00:00:02.38
Pass 1	67	00:00:00.43	00:00:01.29
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	40	00:00:00.31	00:00:00.63
Symbol table output	2	00:00:00.01	00:00:00.01
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	279	00:00:01.32	00:00:04.95

The working set limit was 900 pages.

1789 bytes (4 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 1 non-local and 6 local symbols.

147 source lines were read in Pass 1, producing 8 object records in Pass 2.

0 pages of virtual memory were used to define 0 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

_\$255\$DUA28:[SYSLIB]STARLET.MLB;2

0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/DISA=TRACE/LIS=LISS:MATCHNAME/OBJ=OBJ\$:MATCHNAME MSRC\$:MATCHNAME/UPDATE=(ENHS:MATCHNAME)

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY